MASMOVIL **F** Grafana Faro: RUM with for Apps



















Agenda



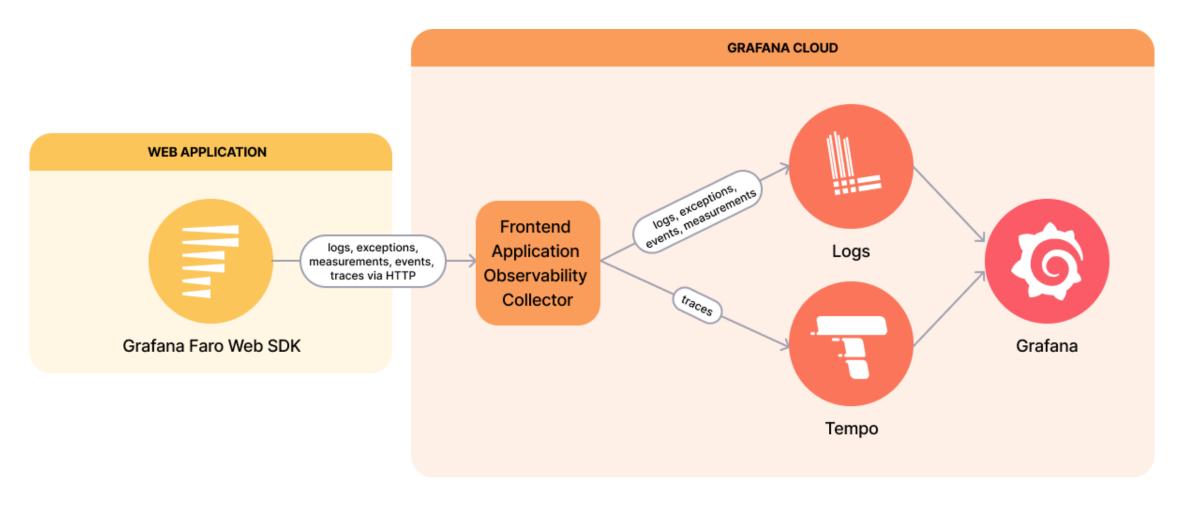
- ¿Qué es y por qué usar Grafana Faro?
- Grafana Faro 101
 - Grafana Faro Web SDK
- Grafana Faro y React
 - ¿Qué le ofrece Grafana Faro a React?
 - ¿Cómo configuramos el uso de Grafana Faro en React?
- Caso de Uso en MásMóvil: Agents
- ¿Qué ventajas nos ofrece Grafana Faro?



¿Qué es Grafana Faro?

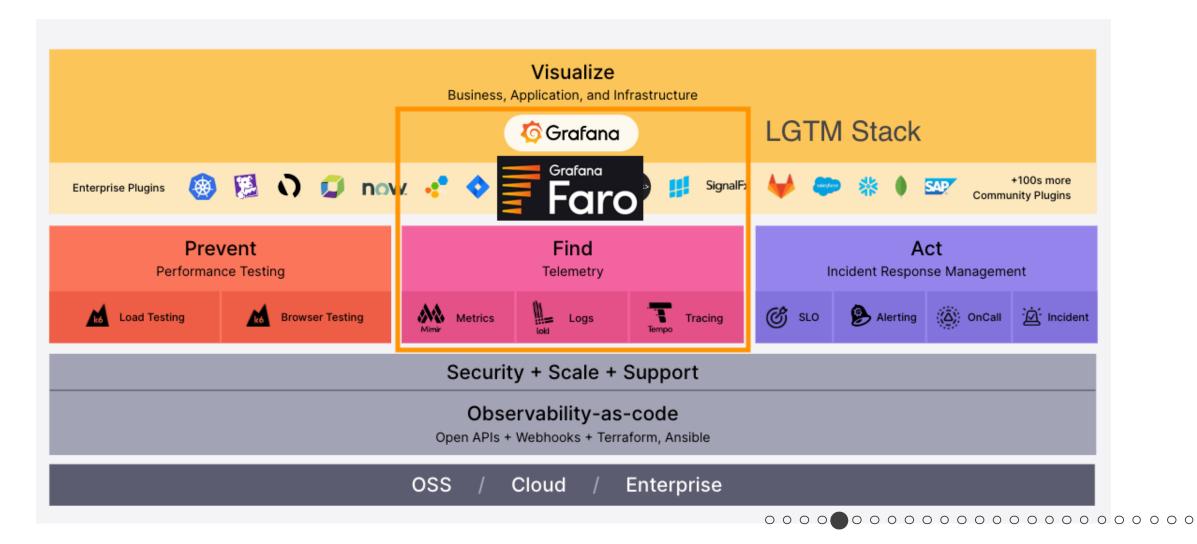
¿Qué es Grafana Faro?







The Open and Composable Observability Stack





¿Por qué usar Grafana Faro?



La observabilidad es





Grafana Faro 101



Grafana Faro Web SDK

¿Qué nos permite observar?









Tracking de errores

Monitorización Web Performance (CWV) Trazas de la pila completa con OpenTelemetry

(errors)

(measurements)

(traces)





Logs de consola para mayor contexto

Tracking de la actividad del usuario

(logs)

(events, session, view, page)

¿Qué herramientas utilizamos?





Instrumentations



tracing con opentelemetry-js



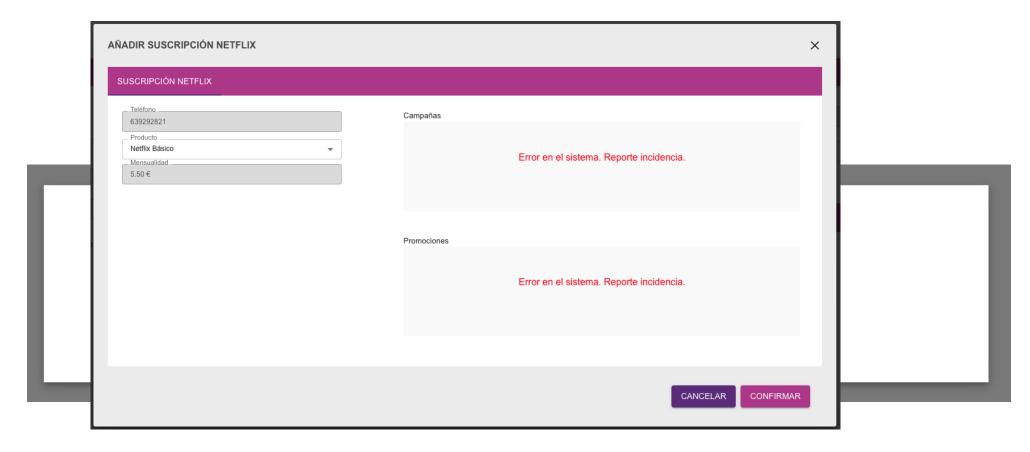
Metas



¿Qué le ofrece Grafana faro a React?









```
1 import { FaroErrorBoundary } from '@grafana/faro-react'
   async function startApp() {
     {...code....}
     const ErrorModal = (error: Error) => {
       return <FallbackModal error={error} />
 8
 9
10
     ReactDOM.render(
       <Suspense fallback={null}>
11
12
         <RootProvider appDeps={appDeps}>
           <FaroErrorBoundary fallback={ErrorModal}>
             <Router history={history}>
14
15
               <App />
             </Router>
17
           </FaroErrorBoundary>
         </RootProvider>
18
19
       </Suspense>,
         document.getElementById('root'),
20
21
22 }
  startApp()
```

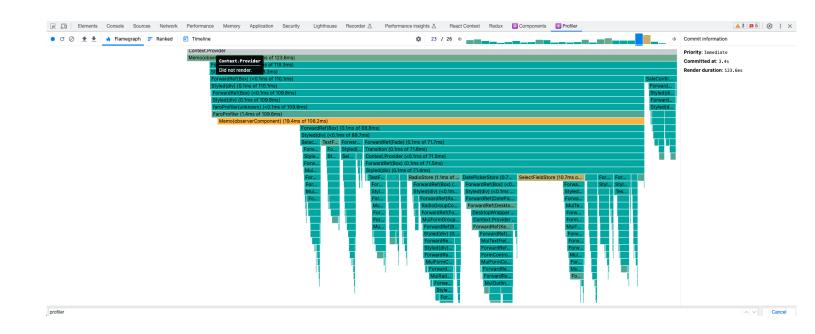


```
1 import { FaroErrorBoundary } from '@grafana/faro-react'
   async function startApp() {
     {...code....}
     const ErrorModal = (error: Error) => {
       return <FallbackModal error={error} />
 8
     ReactDOM.render(
         <RootProvider appDeps={appDeps}>
           <FaroErrorBoundary fallback={ErrorModal}>
             <Router history={history}>
               <App />
             </Router>
         </RootProvider>
         document.getElementById('root'),
23 startApp()
```



```
1 import { FaroErrorBoundary } from '@grafana/faro-react'
   async function startApp() {
     {...code....}
     const ErrorModal = (error: Error) => {
       return <FallbackModal error={error} />
     ReactDOM.render(
         <RootProvider appDeps={appDeps}>
           <FaroErrorBoundary fallback={ErrorModal}>
13
             <Router history={history}>
14
15
               <App />
16
             </Router>
17
           </FaroErrorBoundary>
         </RootProvider>
         document.getElementById('root'),
23 startApp()
```

Analizador de Componente (Component Profiler) MASMOVIL®



Analizador de Componente (Component Profiler) MASMOVIL[®]

```
import { withFaroProfiler } from '@grafana/faro-react';
const ClientForm = observer(() => {
  return (<>{...code....}</>)
})
export default withFaroProfiler(ClientForm);
                                                         • C ⊘ 🖢 🖢 🔒 Flamegraph 🖃 Ranked
                                                                                                                                             Render duration: 123.6ms
```

Analizador de Componente (Component Profiler) MASMOVIL[®]

```
1 import { withFaroProfiler } from '@grafana/faro-react';
  const ClientForm = observer(() => {
    return (<>{...code....}</>)
  export default withFaroProfiler(ClientForm);
                                                   Render duration: 123.6m:
```



Enrutador de páginas (Router)

React Router V6

React Router V5



Enrutador de páginas (Router)

React Router V6

React Router V5



Enrutador de páginas (Router)

React Router V6

React Router V5



Demo de Grafana Faro React





¿Cómo configuramos el uso de Grafana Faro en React?



Inicializando Grafana Faro

Definición del hook con la configuración inicial de Grafana Faro

```
1 import { Route } from 'react-router-dom'
 2 import {
     getWebInstrumentations,
     initializeFaro,
     ReactIntegration,
     ReactRouterVersion,
 7 } from '@grafana/faro-react'
   import { faro } from '@grafana/faro-web-sdk'
   import { TracingInstrumentation } from '@grafana/faro-web-tracing'
10
   import { useAppDepsContext } from '~/...'
12
   export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
15
16
     const faroConfig = () => {
       return initializeFaro({
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
         app: {
```





Definición del hook con la configuración inicial de Grafana Faro

```
1 import { Route } from 'react-router-dom'
     getWebInstrumentations,
     initializeFaro,
     ReactIntegration,
     ReactRouterVersion,
 8 import { faro } from '@grafana/faro-web-sdk'
 9 import { TracingInstrumentation } from '@grafana/faro-web-tracing'
   import { useAppDepsContext } from '~/...'
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
```

```
getWebInstrumentations,
     initializeFaro,
     ReactIntegration,
     ReactRouterVersion,
 8 import { faro } from '@grafana/faro-web-sdk'
9 import { TracingInstrumentation } from '@grafana/faro-web-tracing'
   import { useAppDepsContext } from '~/...'
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
16
     const faroConfig = () => {
       return initializeFaro({
17
18
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
19
         app: {
20
           name: 'agents',
           version: v {runtimeConfig.version},
21
           environment: runtimeConfig.environment,
22
23
         },
         instrumentations: [
24
           ...getWebInstrumentations(),
25
           new TracingInstrumentation(),
26
27
           new ReactIntegration({
28
             router: {
29
               version: ReactRouterVersion.V5,
30
               dependencies: {
```

```
initializeFaro,
     ReactIntegration,
     ReactRouterVersion,
8 import { faro } from '@grafana/faro-web-sdk'
9 import { TracingInstrumentation } from '@grafana/faro-web-tracing'
import { useAppDepsContext } from '~/...'
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
18
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
           version: v {runtimeConfig.version},
           environment: runtimeConfig.environment,
           ...getWebInstrumentations(),
           new TracingInstrumentation(),
           new ReactIntegration({
               version: ReactRouterVersion.V5,
               dependencies: {
                 history,
                 Route,
```

```
ReactIntegration,
     ReactRouterVersion,
 7 } from '@grafana/faro-react'
 8 import { faro } from '@grafana/faro-web-sdk'
9 import { TracingInstrumentation } from '@grafana/faro-web-tracing'
   import { useAppDepsContext } from '~/...'
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
19
         app: {
           name: 'agents',
20
21
           version: v {runtimeConfig.version},
           environment: runtimeConfig.environment,
22
23
         },
           ...getWebInstrumentations(),
           new TracingInstrumentation(),
           new ReactIntegration({
               version: ReactRouterVersion.V5,
                 history,
                 Route,
```

```
import { useAppDepsContext } from '~/...'
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
           version: v {runtimeConfig.version},
           environment: runtimeConfig.environment,
         },
           ...getWebInstrumentations(),
25
           new TracingInstrumentation(),
           new ReactIntegration({
               version: ReactRouterVersion.V5,
                 history,
                 Route,
             },
           }),
       })
```

```
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
           version: v {runtimeConfig.version},
           environment: runtimeConfig.environment,
         },
           ...getWebInstrumentations(),
           new TracingInstrumentation(),
26
           new ReactIntegration({
               version: ReactRouterVersion.V5,
               dependencies: {
                 history,
                 Route,
               },
           }),
         ],
       })
     const faroLog = () => faro.api.pushLog(['Faro was initialized'])
```

```
13 export function useFaroInit() {
     const { runtimeConfig } = useAppDepsContext()
     const faroConfig = () => {
       return initializeFaro({
         url: 'http://faro-collector-us-central-0.grafana.net/collect/{app-key}',
           version: v {runtimeConfig.version},
           environment: runtimeConfig.environment,
         },
         instrumentations: [
           ...getWebInstrumentations(),
           new TracingInstrumentation(),
           new ReactIntegration({
27
28
             router: {
29
               version: ReactRouterVersion.V5,
               dependencies: {
30
31
                 history,
32
                 Route,
33
               },
34
             },
35
           }),
         ],
       })
     const faroLog = () => faro.api.pushLog(['Faro was initialized'])
     return { faroConfig, faroLog, faroApi: faro.api, createSession }
```

```
const faroLog = () => faro.api.pushLog(['Faro was initialized'])
return { faroConfig, faroLog, faroApi: faro.api, createSession }
}
```

Ejecución de la inicialización

```
1 export const Routes = observer(() => {
   useEffect(() => {
      configureAxiosInterceptors()
      faroConfig()
6
      faroLog()
    }, □)
    useEffect(() => {
      if (isAuthenticated && !!agent.id) {
        faroApi.setUser({
        id: agent.id,
          email: agent.email,
          username: agent.userName,
        faroApi.setSession(createSession({ id: agent.id }))
    }, [isAuthenticated, agent])
```

Ejecución de la inicialización



```
1 export const Routes = observer(() => {
     useEffect(() => {
       configureAxiosInterceptors()
       faroConfig()
       faroLog()
     }, □)
     useEffect(() => {
       if (isAuthenticated && !!agent.id) {
         faroApi.setUser({
12
13
          id: agent.id,
           email: agent.email,
14
15
           username: agent.userName,
16
17
         faroApi.setSession(createSession({ id: agent.id }))
     }, [isAuthenticated, agent])
     <Switch>
       </Switch>
26 })
```

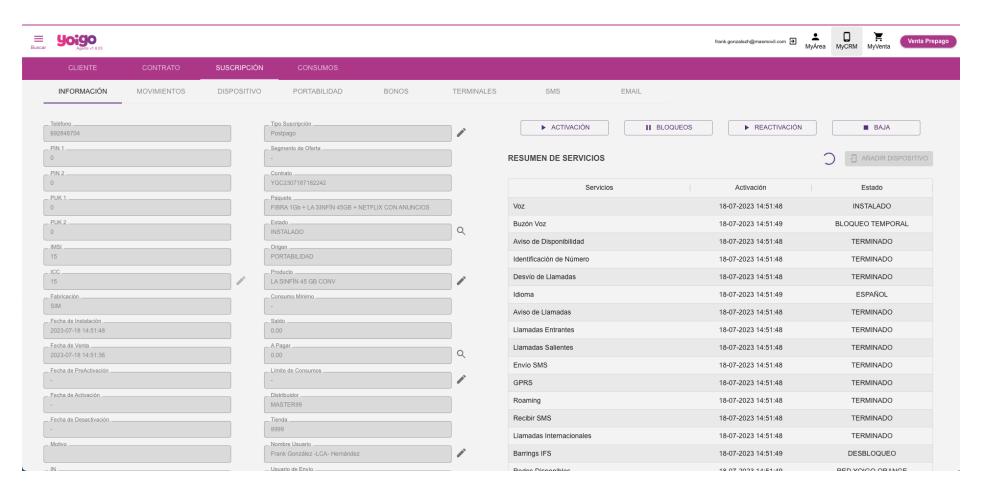


Caso de Uso en MásMóvil: Agents





Portal de Ventas Agents





¿Qué ventajas nos ofrece Grafana Faro?

Ventajas



- Adopción dentro de la cultura tecnológica de la empresa de Grafana Cloud
- Integración y uso de OpenTelemetry
- Tipos de métricas de rendimiento, registros, excepciones, eventos y seguimientos. Ejemplo: Core Web Vitals, User Journey
- Integración con React y uso de componentes adhoc para el manejo de errores, rutas
- Centralización de la monitorización e integración de alertas con prometheus



¿Preguntas?



